(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 363 882 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 89118752.8

(51) Int. Cl.5: G06F 9/46

(22) Date of filing: 09.10.89

(30) Priority: 08.10.88 JP 252904/88
27.10.88 JP 269577/88
27.10.88 JP 269578/88
27.10.88 JP 269579/88
27.10.88 JP 269580/88
27.10.88 JP 269581/88

(43) Date of publication of application:
18.04.90 Bulletin 90/16

(84) Designated Contracting States:
BE DE FR GB IT NL

(71) Applicant: NEC CORPORATION
33-1, Shiba 5-chome
Minato-ku Tokyo(JP)

(72) Inventor: Hayashi, Hideo
c/o NEC Corporation 33-1, Shiba 5-chome
Minato-ku Tokyo(JP)
Inventor: Mochizuki, Atsuo
c/o NEC Koufu, Ltd. 17-14, Marunouchi 1-chome
Koufu-shi Yamanashi(JP)
Inventor: Kobayashi, Ryuji
c/o NEC Software, Ltd. 7-15, Shiba 5-chome
Minato-ku Tokyo(JP)
Inventor: Kumamoto, Chiaki
c/o NEC Software, Ltd. 7-15, Shiba 5-chome
Minato-ku Tokyo(JP)
Inventor: Kokubu, Reiko
c/o NEC Software, Ltd. 7-15, Shiba 5-chome
Minato-ku Tokyo(JP)

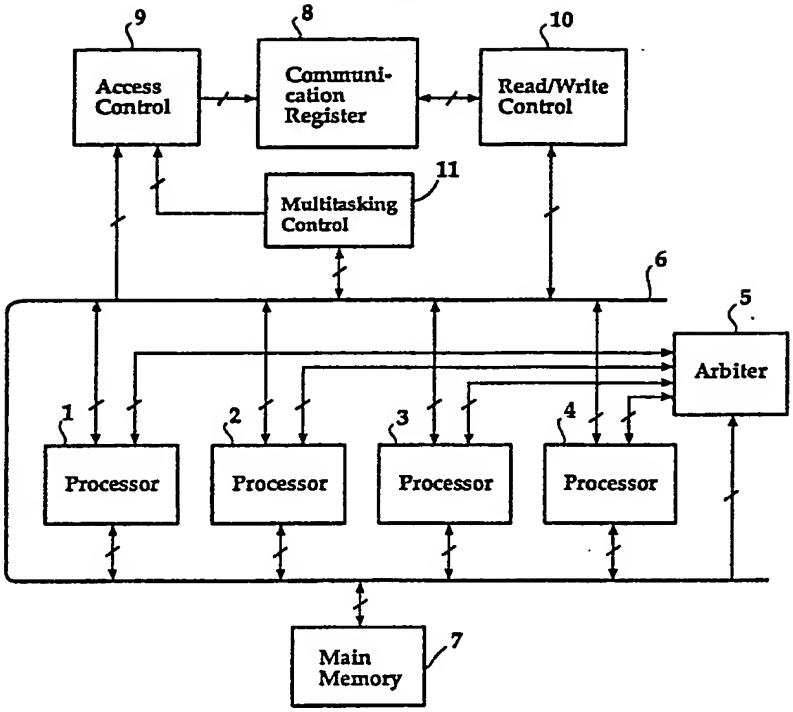
(74) Representative: Vossius & Partner
Siebertstrasse 4 P.O. Box 86 07 67
D-8000 München 86(DE)

(54) Multiprocessor system using communication register having processor-associated storage locations.

(57) In a multiprocessor system, a communication register is partitioned into groups of word storage locations and one of the groups is further partitioned into subgroups associated respectively with the processors. An access controller accesses any groups of the communication register when a system program is being processed and accesses one of the subgroups when a user program is being processed. A write controller is responsive to a test & set instruction of first occurrence from a common bus for assembling a lock word with a data word, a control field and a counter field containing a variable count. The control field of the lock word is set to a first binary state when it is assembled and reset to a second binary state when deassembled. In response to a load instruction from the common bus, either the data word from the bus or lock word is stored into a specified storage area of a communication register. A read controller reads contents of an addressed location of the communication register onto the common bus in response to a save instruction. Test & set instruction of a subsequent occurrence causes the variable count in the stored lock word to be decremented as long as the control field remains set to the first binary state. When the count reduces to zero, a signal is applied to the common bus indicating the occurrence of a dead lock.

EP 0 363 882 A2

Fig. 1


9
8
10
Access Control
Communi-cation Register
Read/Write Control
11
Multitasking Control
6
5
Arbiter
1
2
3
4
Processor
Processor
Processor
Processor
Main Memory
7

Multiprocessor System Using Communication Register Having Processor-Associated Storage Locations

The present invention relates to multiprocessor systems.

The design concept of a multiprocessor system is generally to increase system throughput, but in some cases it is aimed at reducing job's turnaround time. If a job has independent tasks, the turnaround time of the job can be reduced by performing parallel processing on the whole or part of the job. In some type of parallel processing, there is a special kind of data that must be synchronized between tasks or shared by more than one processor. Such data must be used exclusively by one processor. For such purposes, a lock word is stored in a main memory. When a processor attempts to access the exclusive data, it sets the lock word if it is not set so as to obtain the right to access. Any of the other processors accessing the exclusive data examines the status of the lock word, and if it has been set already, the processor is denied access to that data until the lock word is reset. Since the lock word is stored in the main memory, there is an inherent delay in the transfer of data, thus placing a limit to the highest speed with which data can be transferred between processors. In addition to this, a dead lock condition can occur when a processor fails to reset the lock word.

It is therefore an object of the present invention to provide a multiprocessor system which allows high speed data transfer between multiple processors by writing a lock word into a communication register.

Another object of the present invention is to provide a multiprocessor system which eliminates dead lock conditions which might otherwise occur as a result of a processor failing to reset a lock word.

According to a first aspect of the present invention, a communication register is connected to a common bus, the communication register being partitioned into a plurality of groups of word storage locations. One of the groups is further partitioned into a plurality of subgroups associated respectively with the processors. An access controller accesses one of the word storage locations of all groups of the communication register in accordance with a first address code of longer bit length supplied from the common bus when the granted processor is processing a system program and for accessing a storage location of one of the subgroups in accordance with a second address code of shorter bit length supplied from the common bus and contents of directories which are associated respectively with the processors when the granted processor is processing a user program. A write controller writes a data word of longer bit length from the common bus into a word storage location of the communication register specified by the access controller and forms a lock word with a data word of shorter bit length from the common bus and writes the lock word into a specified word storage location of the communication register. A read controller reads out contents of a storage location of the communication register specified by the access controller and forwards them onto the common bus.

According to a second aspect, the access controller accesses a specified storage area of the communication register in accordance with an address code supplied from the common bus. A write controller is responsive to a test & set instruction of first occurrence from the common bus for assembling a lock word with an exclusive data word supplied from the common bus, a control field and a counter field containing a variable count. The control field of the lock word is set to a first binary state when the lock word is assembled and reset to a second binary state when the lock word is deassembled. The write controller is further responsive to a load instruction from the common bus for writing a non-exclusive data word received from the bus as well as the lock word into a storage area of the communication register specified by the address controller. In response to a save instruction, a read controller reads a word out of the communication register onto the common bus. Responsive to the test & set instruction of a subsequent occurrence from the common bus, the count value in the stored lock word is varied by a predetermined amount as long as the control field remains set to the first binary state. A detector is provided to apply a signal to the common bus indicating the occurrence of an abnormal condition when the count value attains a predetermined value.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be described in further detail with reference to the accompanying drawings, in which:

Fig. 1 is a block diagram of a multiprocessor system according to the present invention;

Fig. 2A is a block diagram of the communication register and the address controller of Fig. 1;

Fig. 2B is a block diagram of the commu-

nication register and the read/write controller of Fig. 1;

Fig. 3 shows formats of data stored in the communication register of Fig. 2A;

Fig. 4 shows a format of data stored in a directory control register of the multitasking controller of Fig. 2A;

Figs. 5A and 5B are flowcharts describing a sequence of operations performed by the multitasking controller; and

Fig. 6 is a schematic illustration of the contents of communication register directories used in a multitasking environment.

DETAILED DESCRIPTION

Referring now to Fig. 1, there is shown a multiprocessor system according to an embodiment of the present invention. Four processors 1, 2 3 and 4 are provided. A bus arbiter 5 receives a request from the processors when they attempt to access a common bus 6. Permission is granted to only one processor if there is more than one competing processor. Through the common bus 6 each processor communicates with a main memory 7 and a communication register 8 for exchanging data with other processors. An access controller 9 and a read/write controller 10 are connected to the common bus 6 to access the communication register 8 and write data from a granted processor into a specified word location of the communication register 8 and read it from a specified area of the register. 8. A multitasking controller 11 is further connected between bus 6 and access controller 9 to permit multitasking operations to be performed between the processors in a master-slave relationship.

As illustrated in Fig. 2A, the communication register 8 is partitioned into 256 words of 64 bits each. The register 8 is divided into a group A of #0 to #127 words and a group B of #128 to #255 words. Group B of the register 8 is further divided into subgroups B1 (#128 to #159), B2 (#160 to #191), B3 (#192 to #223) and B4 (#224 to #255).

Access controller 9 comprises a processor identifier register 20, a mode register 21 and a 64-bit address register 22, all of which are connected to the common bus 6 to receive and store data supplied from a processor to which permission is granted from the bus arbiter 5. The identifier number of the accessing processor is stored in the processor identifier register 20, and a mode identifier bit of the processor is stored in the mode register 21. When the processor is performing a system program, it is said to be in a high-priority mode and a logical 1 is stored in the mode register 21 and when performing a user program it is said to be in a low-priority mode and a logical 0 is stored in the register 21. Mode register 21 has output leads 21a and 21b which are connected to adders 29 and 30, respectively. During the high-priority mode, the mode register outputs at leads 21a and 21b are logical 1 and 0, respectively, and during the low-priority mode, the logical levels at leads 21a and 21b are at 0 and 1, respectively.

To permit full access to the whole area of communication register 8 when the system program is being executed, an 8-bit address data is stored in the #56 to #63 bit positions of the 64-bit address register 22 which are connected to adder 29. The output of adder 29 is supplied to the communication register 8 to access a desired word location from any of the 256 word locations of the communication register 8.

Partial access to the register 8 is given to a processor when executing a user program. In this case, a 5-bit address code is stored in the #59 to #63 bit positions of the address register 22 to identify any one of 32 word locations of a subgroup which is addressed. These bit positions are connected to an adder 30 to which the output lead 21b of mode register 21 is also connected. The output of processor identifier register 20 is translated by a decoder 23 into a 4-bit code which is supplied to comparators 24, 25, 26 and 27 for comparison with 4-bit codes supplied respectively from communication register directories 31, 32, 33 and 34 which are, in turn, associated with subgroups B1, B2, B3 and B4 of the communication register 8, respectively.

Each of the directories 31, 32, 33 and 34 has four bit positions. Normally, a logical 1 is set into the first (leftmost) bit position of directory 31, the second bit position of directory 32, the third bit position of directory 33 and the fourth (rightmost) bit position of directory 34 as illustrated in Fig. 2A. The bit position of each directory in which a logical 1 is stored identifies the processor with which the directory is associated, so that directories 31, 32, 33 and 34 are normally associated with processors 1,2,3 and 4, respectively. Since directories 31 to 34 are associated with subgroups B1 to B4, respectively, processors 1 to 4 are normally associated with subgroups B1 to B4, respectively.

Each of the comparators 24 to 27 produces a logical-1 output when there is a match between the input from the decoder 23 and the input from the associated directory and produces a logical 0 output when there is no match between them. A 4-bit code is thus formed by the outputs of comparators 24 to 27 and translated by an encoder 28 into a 2-bit subgroup identifying code, which is applied to the adder 30 and appended with the 5-bit word identifying code to access any location of a speci-

fied one of subgroups B1 through B4. To the leftmost position of the 5-bit code is appended a logical 1 which is supplied from the mode register 21 during a low-priority mode. With a word location of the communication register 6 being identified by the access controller 9, a 64-bit word can be exchanged between the processor and the communication register 8 and this operation is handled by the read/write controller 10 of Fig. 2B.

Read/write controller 10 includes an instruction register 40 and a 64-bit input data register 41 both of which are connected to the bus 6. Instruction register 40 receives three types of instruction from the processors, which are SAVE, LOAD and TEST & SET instructions. Input data register 41 receives a 64-bit data word of a 32-bit "exclusive" data word from the bus 6. An output data register 46 is connected to the communication register 8 to store a data word or a lock word and transfers them to the bus 6 in response to an output from an OR gate 47 to which SAVE and TEST & SET instructions are supplied from a decoder 42.

Data stored in the instruction register 40 is examined by decoder 42 to selectively control the data stored in the input data register 41, depending on whether the instruction is LOAD or TEST & SET as well as on the logical state of the #0 bis position of the output data register 46.

The #0 bit position of the output data register 46 is connected to a zero detector 49 as well as to an AND gate 43 and an inverter 44, and further to the #0 bit position of the input of communication register 8. The #8 to #31 bit positions of the output data register 46 are connected to a decrement circuit 48 whose outputs are connected to the #8 to #31 bit positions of the input of communication register 8 as well as to the zero detector 49. The #32 to #63 bit positions of the output data register 46 are connected to the corresponding bit positions of communication register 8.

Each of the 64-bit words stored in the communication register 8 may be of a 64-bit length data word format as shown in part (a) of Fig. 3, or of a lock word format as shown in part (b) of Fig. 3. In the former case, the communication register 8 is used as a data register and in the latter case, the data word is partitioned into a lock/unlock bit position (#0), a 24-bit counter field (#8 to #31 bit positions) and a 32-bit data field (#32 to #63 bit positions). The lock/unlock bit is set to logical 1 when exclusive control is being carried out and is reset to logical 0 when exclusive control is released.

If the input data is a 64-bit data word, it is supplied to the communication register 8 and if it is a 32-bit "exclusive" data word, a lock/unlock bit and a counter field are appended to it to form a lock word. In response to a LOAD instruction the contents of the input data register 41 are loaded into a word location of the communication register 8 which is specified by the access controller 9 in a manner as described previously.

If a logical 0 is reset in the #0 bit position of a lock word, a logical 0 appears in the #0 bit position of the output data register 46. A TEST & SET instruction causes the output data register 46 to read the stored lock word, so that AND gate 45 is activated to set a logical 1 into the #0 position of the input of communication register 8, and set all 1's into the #8 to #31 bit positions of the register 8 input. In this instance, the #32 to #63 bit input positions are filled with data bits of a 32-bit "exclusive" data word. In this way, a new lock word is assembled with a counter field set with a maximum count value. In response to a TEST & SET instruction, this lock word is stored into a specified location of the communication register 8. The processor, which has issued this TEST & SET instruction, receives this lock word and checks its #0 bit position in response to receipt of a subsequent command to see whether the attempt to set a lock word has been successful or failed. If the attempt has failed, the processor generates a TESt & SET instruction again to repeat the above process until the stored lock word is reset by another processor which has set it previously.

With a logical 1 being set in the #0 bit position of a lock word, a logical 1 appears in the #0 bit position of output data register 46. A subsequent TEST & SET instruction from a processor causes the output data register 46 to read a stored lock word from register 8 and deactivates AND gate 45 and activates AND gate 43. As a result, the count value set in the counter field (#8 to #31 bit positions) of the lock word is decremented by one in the decrement circuit 48, and communication register 8 receives the outputs of decrement circuit 48 and an "exclusive" data word from the #32 to #63 bit positions of output data register 46. The contents of the output data register 46 are therefore updated and stored again into the communication register 8. It is seen therefore that the value set in the counter field of a lock word is reduced to zero when it has been decremented 2^{24} times from the initial value each time the communication register 8 is accessed with a TEST & SET instruction.

In response to a SAVE instruction, the output data register 46 transfers the contents of the output data register 46 to the bus 6.

Zero detector 49 is enabled in response to the output of AND gate 43 to supply a signal to the bus 6 indicating the occurrence of a deadlock condition when the logical value decremented by the decrement circuit 48 is reduced to zero. When this occurs, the processor of interest enters a subroutine to remove the deadlock condition.

The contents of communication register directories 31 through 34 are controlled by a 64-bit directory control instruction supplied from a processor to a directory control register contained in the multitasking controller 11 as shown in Fig. 4. The directory control instruction data includes a control bit in the #55 bit position and master processor identifiers in the #60 to #63 bit positions associated respectively with processors 1, 2, 3 and 4. A logical 1 in the #60 to #63 bit positions of the directory control data indicates that a master processor is one that is associated with that bit location. As shown in Figs. 5A and 5B, multitasking controller 11 is programmed to perform directories set and reset operations.

In Fig. 5A, the program starts with decision block 50 which examines the control bit (C) to check to see if it is logical 1 or 0. If C = 1, exit is to decision block 51 to determine which one of the processors has been granted access to the communication register 8. Depending on the identified processor number, control exits to one of operations blocks 52 to 55 to reset a bit position of a respective one of the directories 31 to 34. Following blocks 52 to 55, exit is to one of blocks 56 to 59 associated respectively with blocks 52 to 55 to set a variable *n* with a processor identifier detected by block 51. Control then exits to decision block 60 to detect which one of the #60 to #63 bit positions of the directory control data is set with a logical 1. Depending on the decision made by block 60, control proceeds to one of operations blocks 61 to 64 to set the #n bit position of a respective one of the directories.

If, for example, processor 3 acts as a slave processor performing a multitasking operation with processor 2 which acts as a master processor, the control bit of a directory set instruction from processor 3 is set to 1 as shown in Fig. 6, and control exits from block 51 to block 54 to reset the #3 bit position of directory 33 to 0 as indicated by a dotted line 33a in Fig. 6. Control advances to block 58 to set *n* to 3. Since processor 2 is the master processor, a 1 is set in the #61 bit position of the directory control data and control exits to block 62 to set a 1 into #2 bit position of directory 32 as indicated by a dotted line 32a. In this way, the #2 and #3 bit positions of directory 32 are set with a logical 1. The contents of these bit positions are accessed by processors 2 and 3 to perform a multitasking operation.

When resetting the contents of directory 32, a directory reset instruction containing C = 0 is supplied from processor 3 to the directory control register of multitasking controller 11 as shown in Fig. 6. In Fig. 5A, the program starts with decision block 65 after checking that C = 0. Exit then is to decision block 66 (Fig. 5B) to determine whether the variable *n* is 1,2,3 or 4. Depending on the variable *n*, control exits to one of operations blocks 67 to 70 to set a bit position of a respective one of the directories 31 to 34. Following blocks 67 to 70, exit is to decision block 71 to detect which one of the #60 to #63 bit positions of the directory reset instruction is set with a logical 1. Depending on the decision made by block 60, control proceeds to one of operations blocks 72 to 75 to reset the #n bit position of a respective one of the directories.

Therefore, in response to the directory reset instruction from processor 3, control exits from block 66 to block 68 to set the #3 bit position of directory 33 to 1 as indicated by a dotted line 33b. Control advances to block 73 to reset #2 bit position of directory 32 to 0 as indicated by a dotted line 32b. In this way, directories 32 and 33 are reset to normal to terminate the multitasking operation.

The foregoing description shows only one preferred embodiment of the present invention. Various modifications are apparent to those skilled in the art without departing from the scope of the present invention which is only limited by the appended claims. Therefore, the embodiment shown and described is only illustrative, not restrictive.

Claims

1. A multiprocessor system having a plurality of processors connected to a common bus, means for granting permission to one of said processors to access said common bus, and a main memory connected to said common bus for storing data to be exchanged between said processors, comprising:

a communication register connected to said common bus, said communication register being partitioned into a plurality of groups of word storage locations, one of said groups being further partitioned into a plurality of subgroups associated respectively with said processors;

first access control means for accessing one of the word storage locations of said first and second groups of said communication register in accordance with a first address code of longer bit length supplied from said common bus when the granted processor is operating in a first mode;

a plurality of directories associated respectively with said subgroups, each of said directories having a plurality of bit positions associated respectively with said processors;

second access control means for accessing one of the word storage locations of one of said subgroups in accordance with a second address code of shorter bit length supplied from said common bus and contents of the bit positions of said direc-

tories when the granted processor is operating in a second mode;

write control means for writing a data word of longer bit length from said common bus into a word storage location of the communication register either specified by the first or second access control means, forming a lock word with a data word of shorter bit length from said bus and writing the lock word into the specified word storage location of the communication register; and

read control means for reading contents of a word storage location of said communication register specified by either of said first and second access control means onto said common bus.

2. A multiprocessor system as claimed in claim 1, further comprising multitasking control means for rewriting the contents of said directories in accordance with a directory control instruction from said granted processor when same is performing a multitasking operation with a master processor indicated by said directory control information.

3. A multiprocessor system as claimed in claim 2, wherein said directories are organized into a matrix of rows and columns, both of said rows and columns being respectively associated with said processors, the intersections of said rows and columns corresponding to the bit positions of said directories, wherein said multitasking control means rewrites the bit position of a directory at the intersection of the row and column which are associated with the granted processor, and rewrites the bit position of a directory at the intersection of the row which is associated with said master processor and the column which is associated with said granted processor.

4. A multiprocessor system as claimed in any one of claims 1 to 3, wherein said write control means is responsive to a test & set instruction of first occurrence from said common bus for assembling said lock word with said data word of shorter bit length, a control field and a counter field containing a variable count, said control field being set to a first binary state when said lock word is assembled and reset to a second binary state when said lock word is deassembled, said write control means being further responsive to a load instruction from said common bus for writing said lock word into a word storage area of said communication register specified by either of said first and second address control means;

means responsive to the test & set instruction of a subsequent occurrence from said common bus for varying said variable count in the stored lock word by a predetermined amount when said control field remains set to said first binary state; and

means for applying a signal to said common bus indicating the occurrence of an abnormal condition when said variable count attains a predetermined value.

5. A multiprocessor system as claimed in any one of claims 1 to 4, wherein said write control means comprises:

input means connected to said common bus for receiving said data words of longer or shorter bit length; and

a logic circuit responsive to said test & set instruction for storing a predetermined bit into a first bit position of said input means and all 1's into second bit positions of the input means, and responsive to said load instruction for loading the contents of said input means into a specified storage word location of said communication register, said input means containing said data word in third bit positions thereof, wherein said read control means comprises:

output means connected to said communication register for receiving contents of a specified word storage location of said communication register in response to each of said test & set instruction and said save instruction, said output means having a first bit position connected to the first bit position of said input means, second bit positions connected to said varying means and third bit positions connected to the third bit positions of said input means, all bit positions of said output means being connected to said common bus,

said logical circuit being responsive to the predetermined bit in said first bit position of said output means for rewriting the second bit positions of said input means with outputs of said varying means and rewriting the third bit positions of said input means with contents of the third bit positions of said output means.

6. A multiprocessor system having a plurality of processors connected to a common bus, means for granting permission to one of said processors to access said common bus, and a main memory connected to said common bus for storing data to be exchanged between said processors, comprising:

a communication register connected to said common bus;

access control means for accessing a specified storage area of said communication register in accordance with an address code supplied from said common bus;

write control means responsive to a test & set instruction of first occurrence from said common bus for assembling a lock word with an exclusive data word supplied from said common bus, a control field and a counter field containing a variable count, said control field being set to a first binary state when said lock word is assembled and reset to a second binary state when said lock word is deassembled, said write control means being further responsive to a load instruction from said

common bus for writing a non-exclusive data word from said bus and said lock word into a storage area of said communication register specified by said address control means;
read control means for reading a word from said communication register onto said common bus in response to a save instruction therefrom;
means responsive to the test & set instruction of a subsequent occurrence from said common bus for varying said variable count in the stored lock word by a predetermined amount when said control field remains set to said first binary state; and
means for applying a signal to said common bus indicating the occurrence of an abnormal condition when said variable count attains a predetermined value.

7. A multiprocessor system as claimed in claim 6, wherein said communication register is partitioned into first and second groups of equal size and said second group is further partitioned into a plurality of equal subgroups respectively associated with said processors, wherein said processors operate in one of first and second modes, wherein said access control means accesses any of said first and second groups when the granted processor is operating in said first mode and accesses one of said subgroups which is associated with said granted processor when same is operating in said second mode.

8. A multiprocessor system as claimed in claim 7, wherein said access control means further comprises:
a plurality of directories associated respectively with said subgroups, each of said directories having a plurality of bit positions associated respectively with said processors; and
means for addressing one of said subgroups in accordance with an address code supplied from said granted processor and contents of the bit positions of said directories when the granted processor is operating in said second mode.

9. A multiprocessor system as claimed in claim 8, further comprising multitasking control means for rewriting the contents of said directories in accordance with a directory control instruction from said granted processor when same is performing a multitasking operation with a master processor indicated by said directory control information.

10. A multiprocessor system as claimed in claim 9, wherein said directories are organized into a matrix of rows and columns, both of said rows and columns being respectively associated with said processors, the intersections of said rows and columns corresponding to the bit positions of said directories, wherein said multitasking control means rewrites the bit position of a directory at the intersection of the row and column which are associated with the granted processor, and rewrites the bit position of a directory at the intersection of the row which is associated with said master processor and the column which is associated with said granted processor.

11. A multiprocessor system as claimed in any one of claims 6 to 10, wherein said write control means comprises:
input means connected to said common bus and said communication register; and
a logic circuit responsive to said test & set instruction for storing a predetermined bit into a first bit position of said input means and all 1's into second bit positions of the input means, and responsive to said load instruction for loading the contents of said input means into a specified area of said communication register, said input means storing said exclusive data word in third bit positions thereof, wherein said read control means comprises:
output means connected to said communication register for receiving contents of a specified area of said communication register in response to either of said test & set instruction or said save instruction, said output means having a first bit position connected to the first bit position of said input means, second bit positions connected to said varying means and third bit positions connected to the third bit positions of said input means, all bit positions of said output means being connected to said common bus,
said logical circuit being responsive to the predetermined bit in said first bit position of said output means for rewriting the second bit positions of said input means with outputs of said varying means and rewriting the third bit positions of said input means with contents of the third bit positions of said output means.

Fig. 1

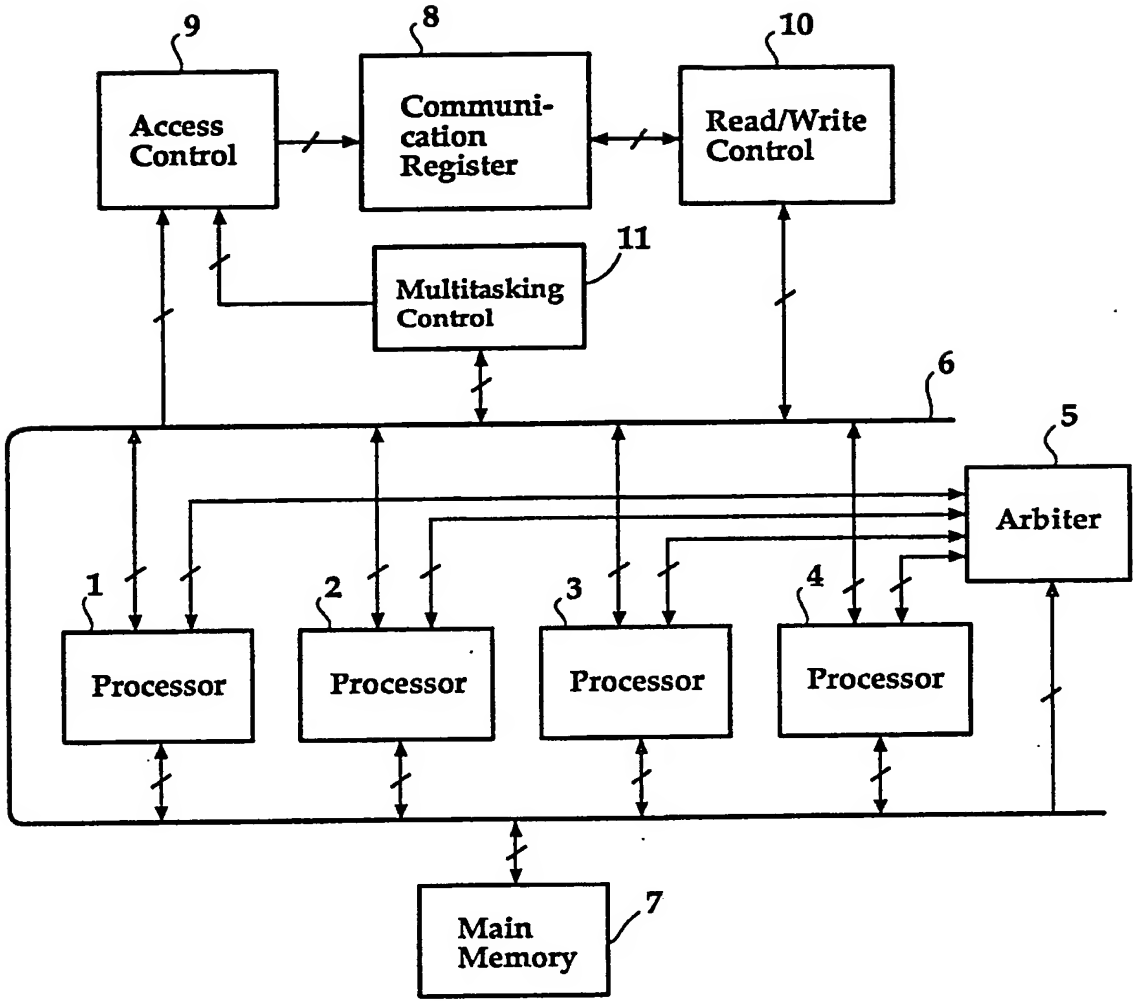

9
8
10
Access Control
Communi-cation Register
Read/Write Control
11
Multitasking Control
6
5
Arbiter
1
2
3
4
Processor
Processor
Processor
Processor
Main Memory
7

Fig. 3

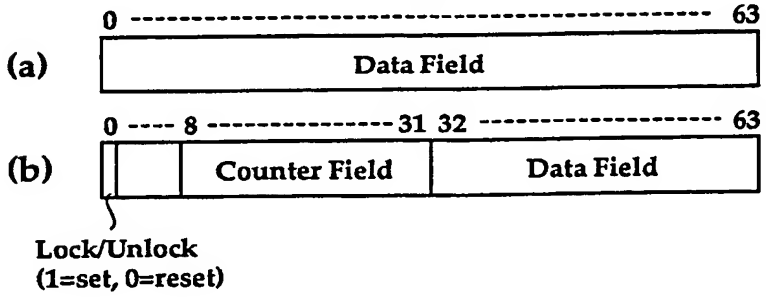
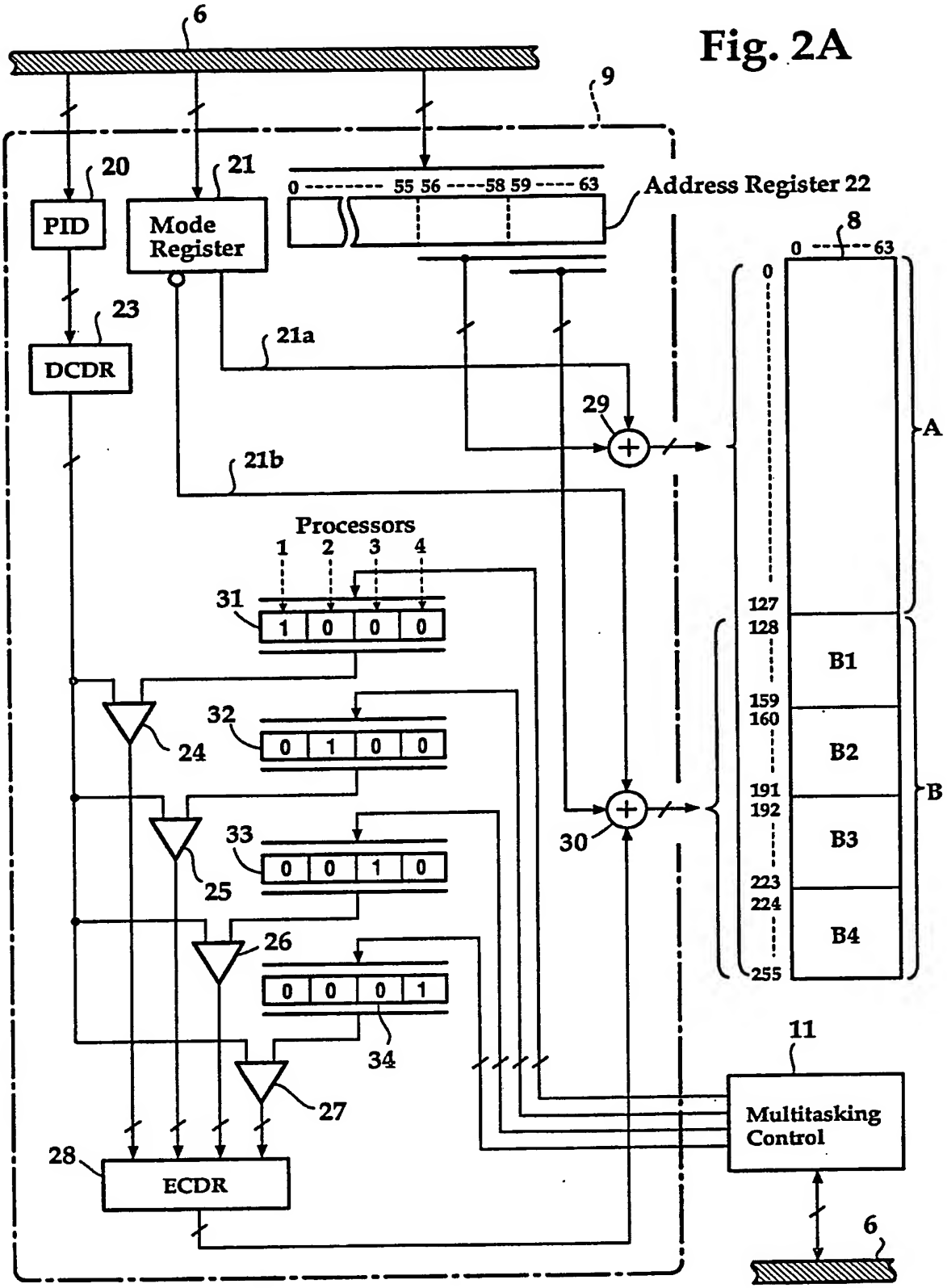

0
63
(a)
Data Field
0
8
31
32
63
(b)
Counter Field
Data Field
Lock/Unlock
(1=set, 0=reset)

Fig. 2A


6
9
20
21
PID
Mode Register
0 ---------- 55 56 ----58 59 ---- 63
Address Register 22
8
0 ---- 63
23
DCDR
21a
29
21b
A
Processors
1 2 3 4
31
1 0 0 0
127
128
B1
159
160
B2
191
192
B3
223
224
B4
255
B
32
24
0 1 0 0
30
33
25
0 0 1 0
26
0 0 0 1
34
11
27
Multitasking Control
28
ECDR
6

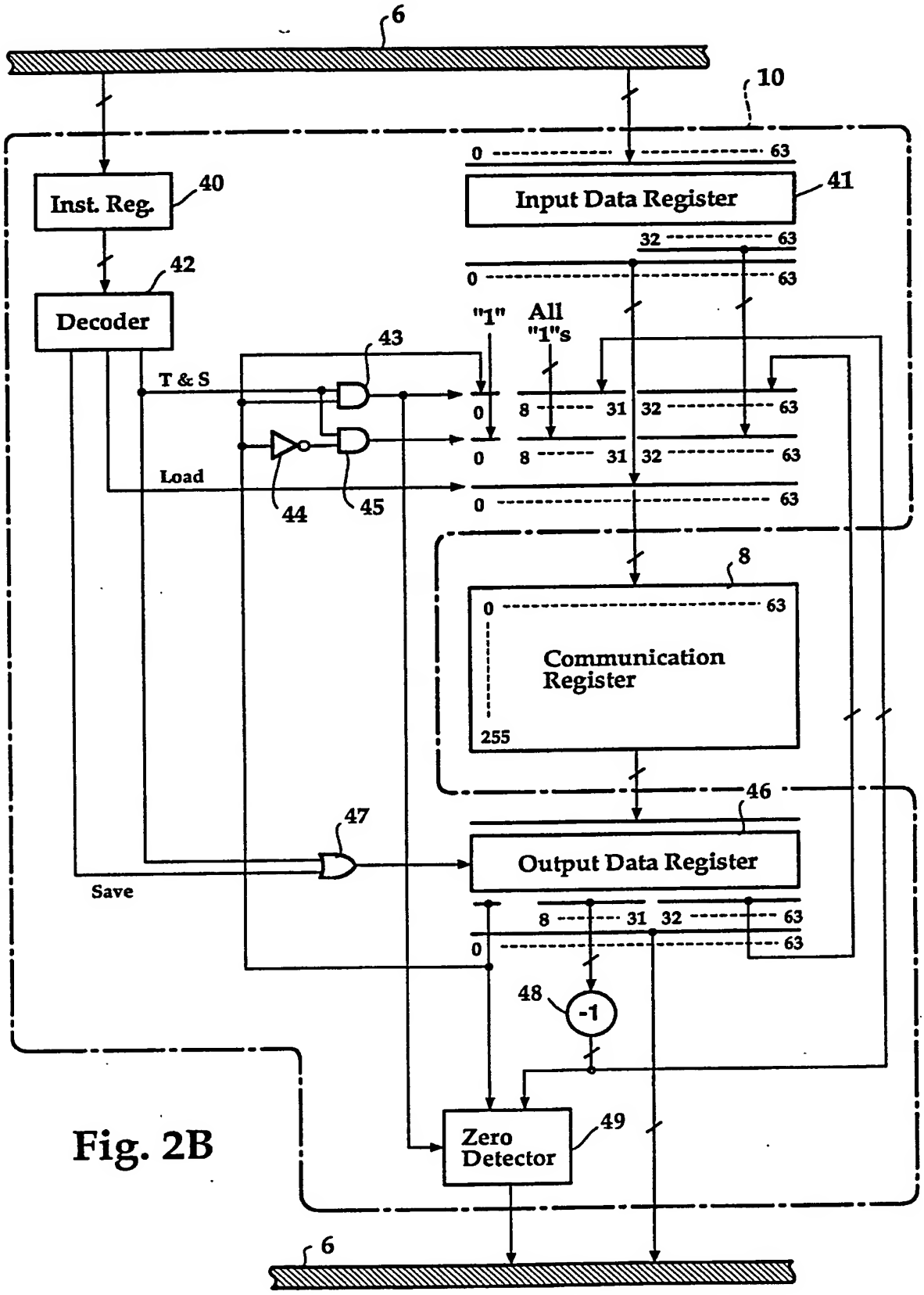

6
10
Inst. Reg.
40
Input Data Register
41
Decoder
42
"1"
All "1"s
43
T & S
Load
44
45
8
Communication Register
255
46
47
Output Data Register
Save
48
-1
Zero Detector
49
6

Fig. 2B

Fig. 4

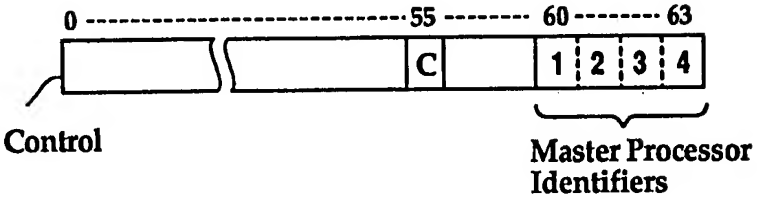
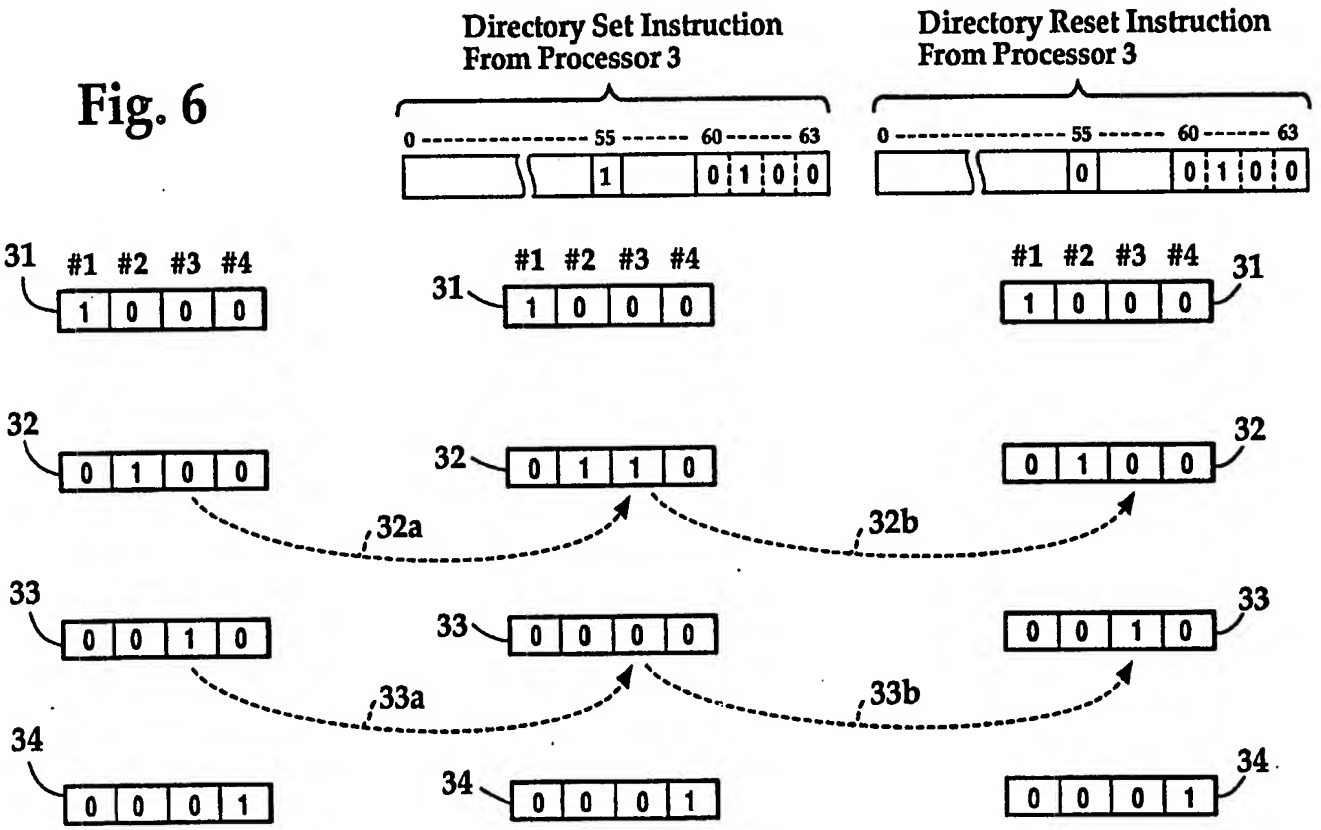


Fig. 6



EP 0 363 882 A2

Fig. 5A

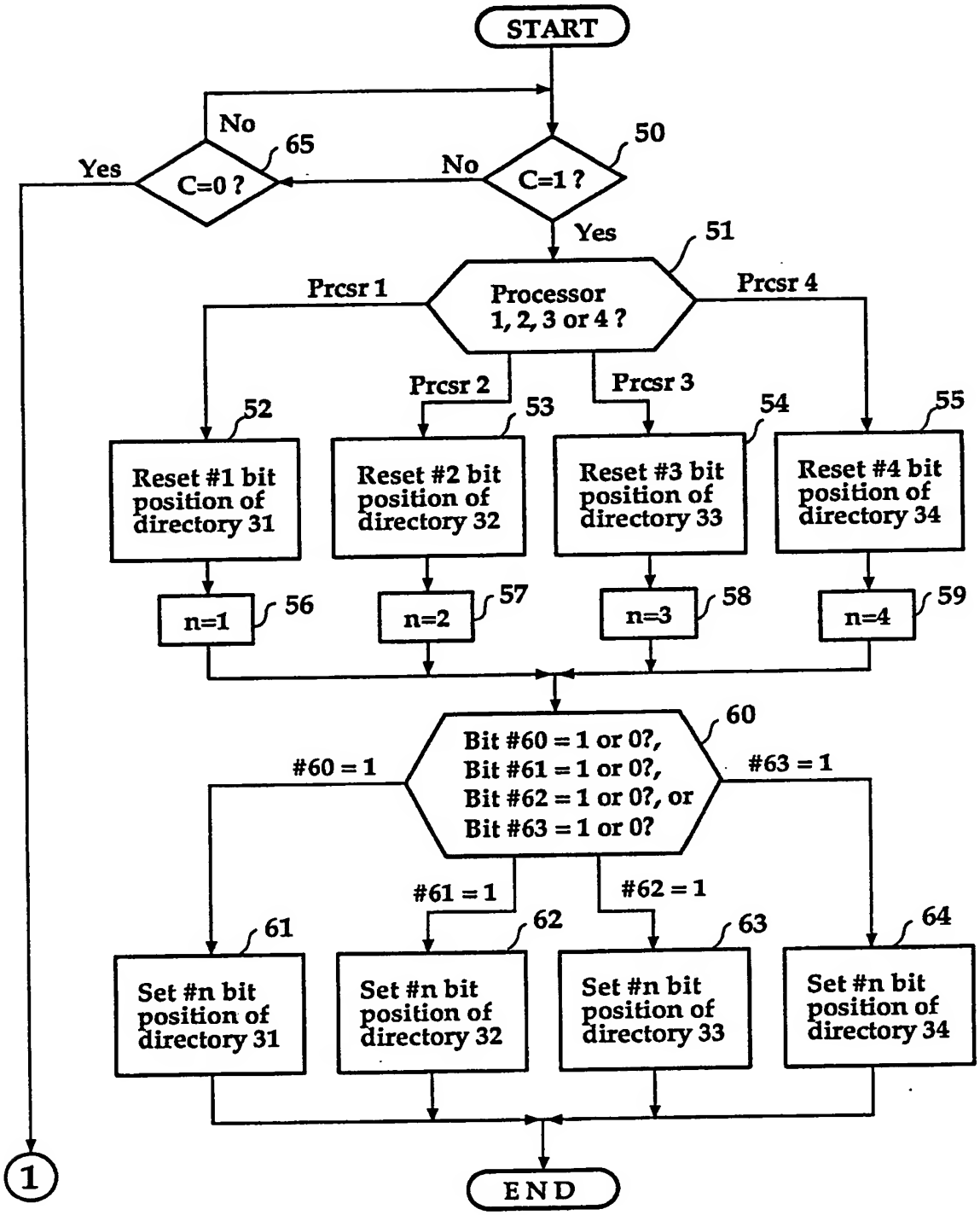
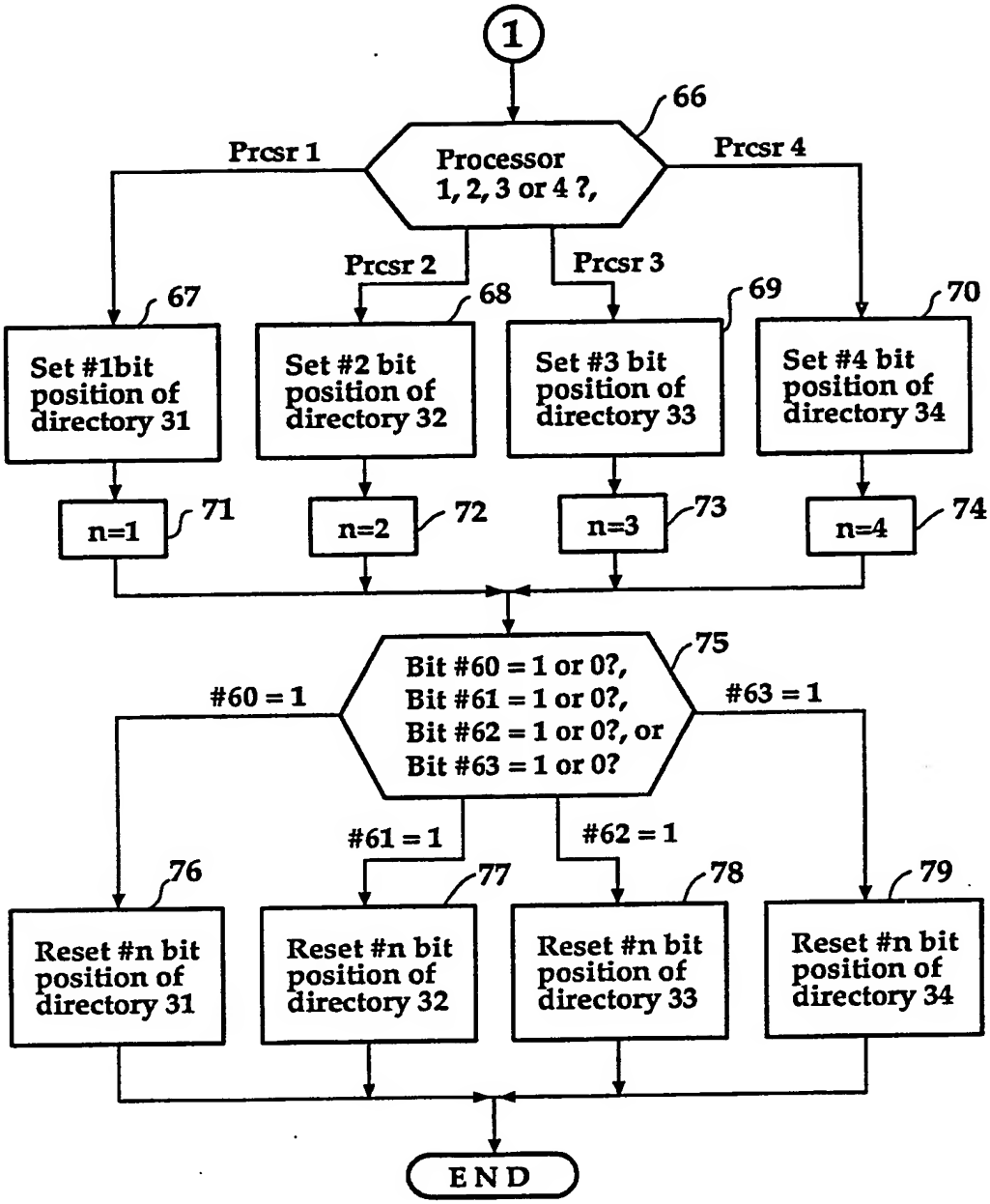

START
No
65
Yes
C=0 ?
No
C=1 ?
50
Yes
51
Prcsr 1
Processor
1, 2, 3 or 4 ?
Prcsr 4
Prcsr 2
Prcsr 3
52
53
54
55
Reset #1 bit position of directory 31
Reset #2 bit position of directory 32
Reset #3 bit position of directory 33
Reset #4 bit position of directory 34
n=1
56
n=2
57
n=3
58
n=4
59
60
Bit #60 = 1 or 0?,
Bit #61 = 1 or 0?,
Bit #62 = 1 or 0?, or
Bit #63 = 1 or 0?
#60 = 1
#63 = 1
#61 = 1
#62 = 1
61
62
63
64
Set #n bit position of directory 31
Set #n bit position of directory 32
Set #n bit position of directory 33
Set #n bit position of directory 34
END
1

Fig. 5B


1
66
Processor 1, 2, 3 or 4 ?,
Prcsr 1
Prcsr 2
Prcsr 3
Prcsr 4
67
Set #1bit position of directory 31
68
Set #2 bit position of directory 32
69
Set #3 bit position of directory 33
70
Set #4 bit position of directory 34
n=1
71
n=2
72
n=3
73
n=4
74
75
Bit #60 = 1 or 0?, Bit #61 = 1 or 0?, Bit #62 = 1 or 0?, or Bit #63 = 1 or 0?
#60 = 1
#61 = 1
#62 = 1
#63 = 1
76
Reset #n bit position of directory 31
77
Reset #n bit position of directory 32
78
Reset #n bit position of directory 33
79
Reset #n bit position of directory 34
END

(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **04070439 A**
(43) Date of publication of application: **05.03.1992**

(51) Int. Cl **E04B 1/24**
E04C 3/32

(21) Application number: **02184827**
(22) Date of filing: **11.07.1990**

(71) Applicant: **NATL HOUSE IND CO LTD**
(72) Inventor: **YAMAOKA MASAKI**

(54) **PILLAR BODY FOR MOUNTING BEAM**

(57) Abstract:

PURPOSE: To obviate the welding work in the site so as to enhance the working efficiency by protruding a bolt shaft to a pillar base body, formed of a short scale angle steel pipe, from the reverse side through a back board material and also screwing a beam mounting metal fixture thereafter fixing a beam.

CONSTITUTION: A plurality of bolt shafts 6 are protruded at a right angle to a pillar base body 2, consisting of a short scale angle steel pipe of sectional shape equal to a pillar body 1, from each surface in the reverse side through a back board material of width not interfering with each other. Next, a beam mounting member 32 is screwed to the pillar base body 2 by bolts 6 while fixing the pillar base body 2 to a beam mounting position of the pillar body 1. Further, a beam D is fixed by bolts to the beam mounting metal fixture 32. In a work spot, assembly is performed without performing welding work. In this way, efficiency of work can be improved.